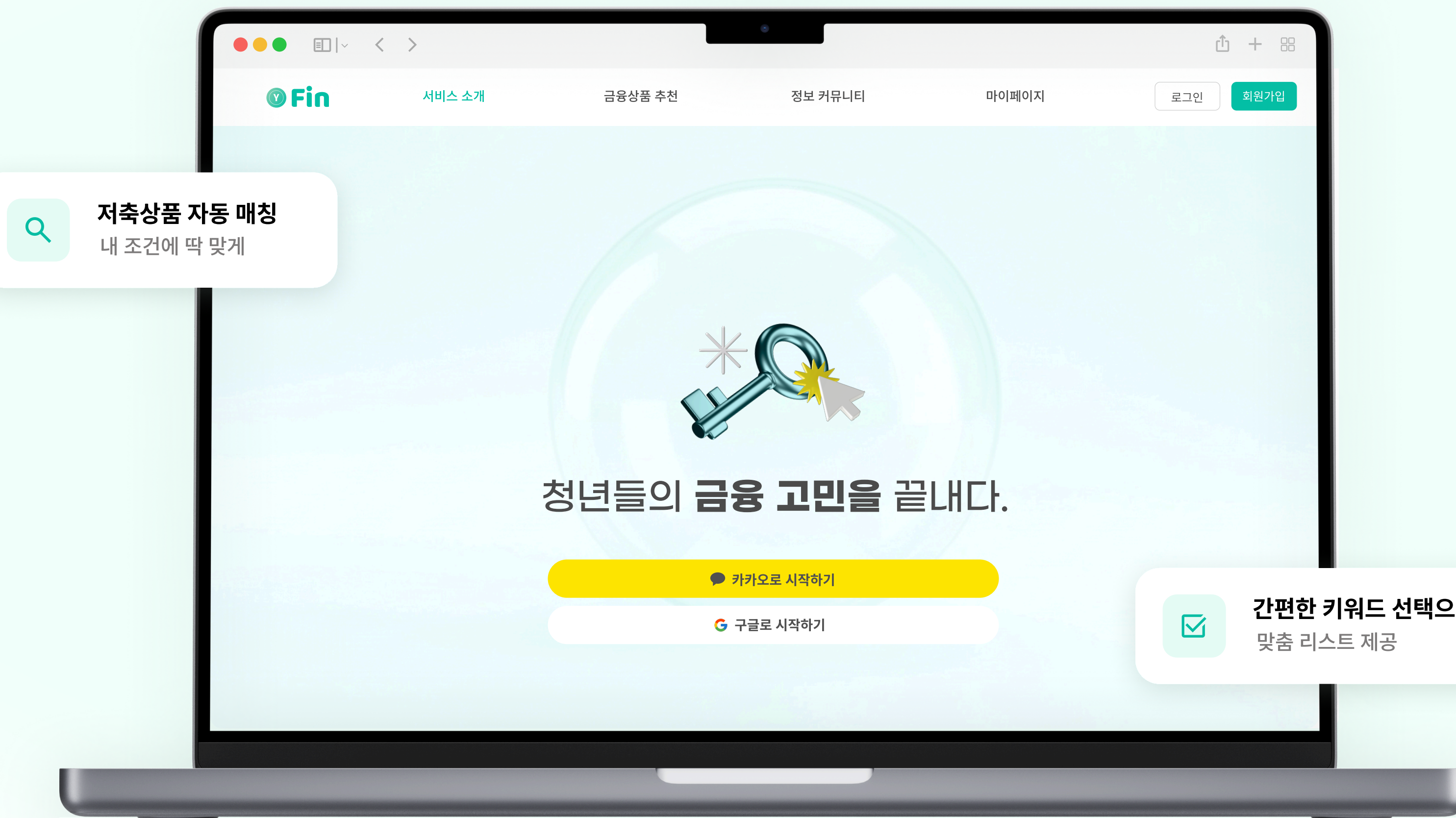


YOUTH · FINANCE · FINISH

청년들의 금융 고민, Y-Fin. 이 끝내다.

금융초보 청년을 위한 개인 맞춤 저축상품 추천 서비스



AGENDA

발표 목차

1 시장 조사 및 문제 정의

시장 조사 · 문제 정의

2 솔루션 및 차별점

3단계 추천 로직 · 적합도 / 세후 실수령액 · 포지셔닝

3 서비스 진행 현황

서비스 시연 영상 · 개발 현황 · AI 활용

4 서비스 향후 계획

상품군 / 기능 확장 · 시장 진입 로드맵 · KPI · 사회적 가치

01

시장조사 및 문제정의

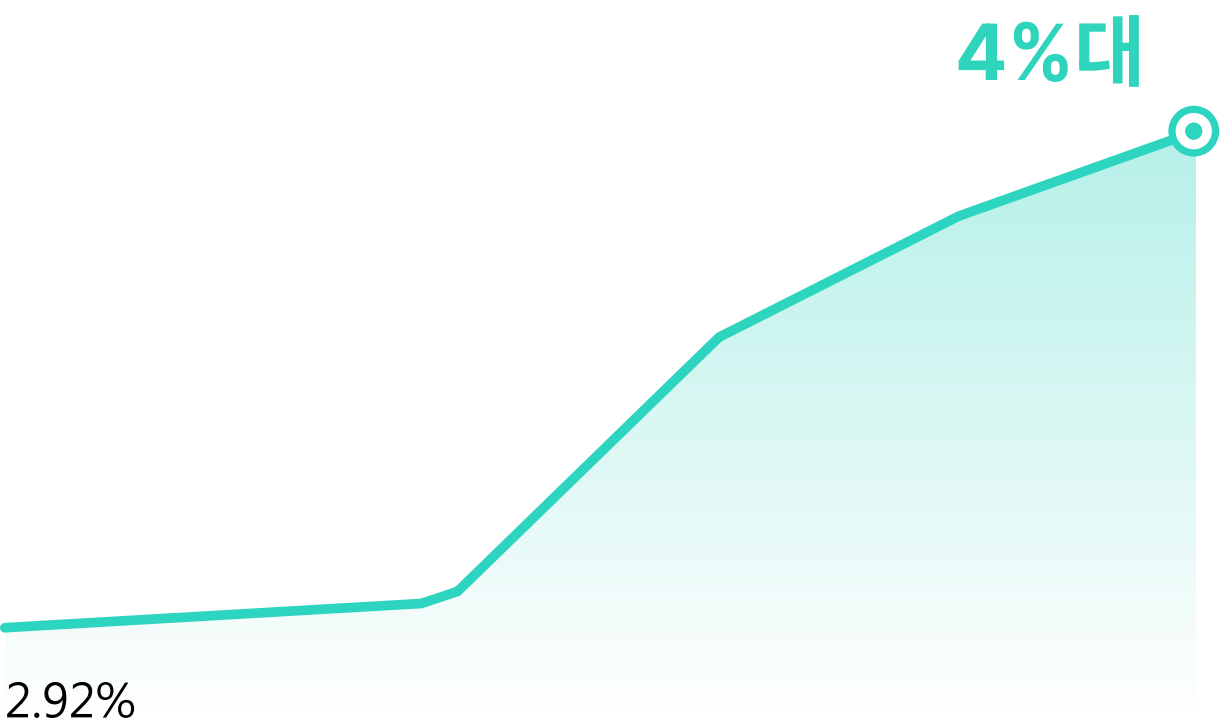
— 시장 조사 · 문제 정의

01 시장 조사

금리·정책·수요 3축 동시 상승, 구조적인 진입 적기

금리

상승 꺾

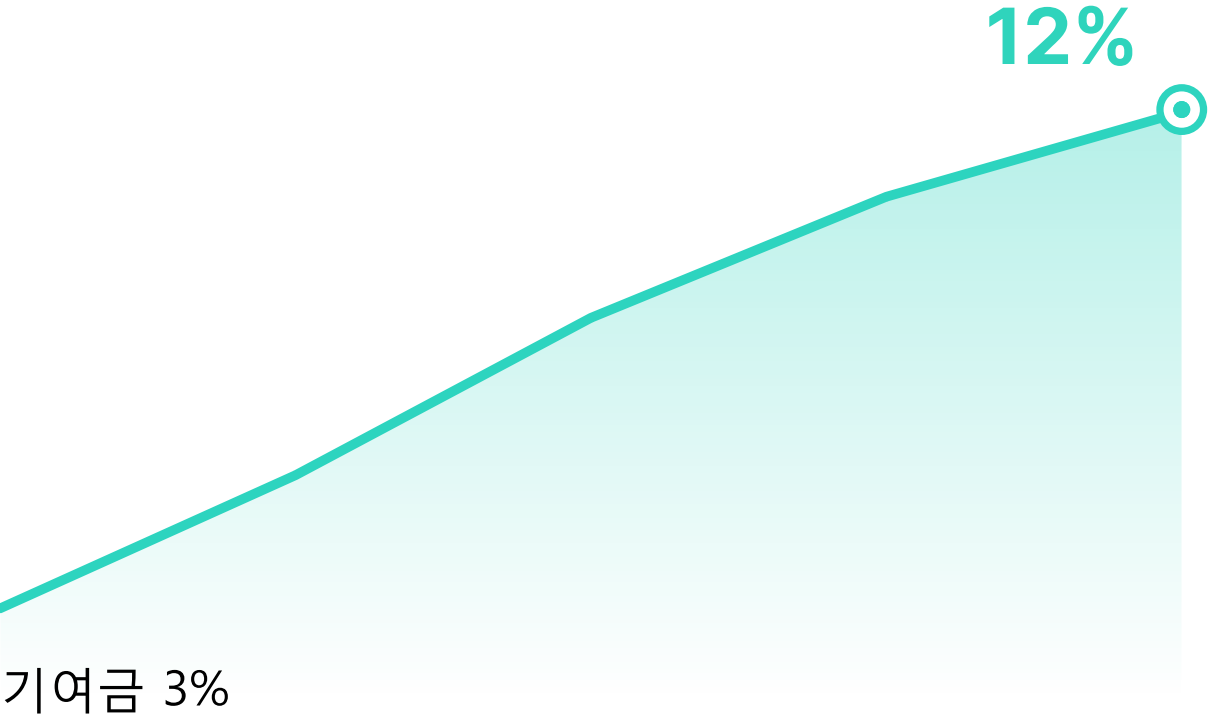


연 4%대

저축은행 예금금리 2.92% → 3.74%,
인터넷·지방은행 특판 재등장

정책

상승 꺾

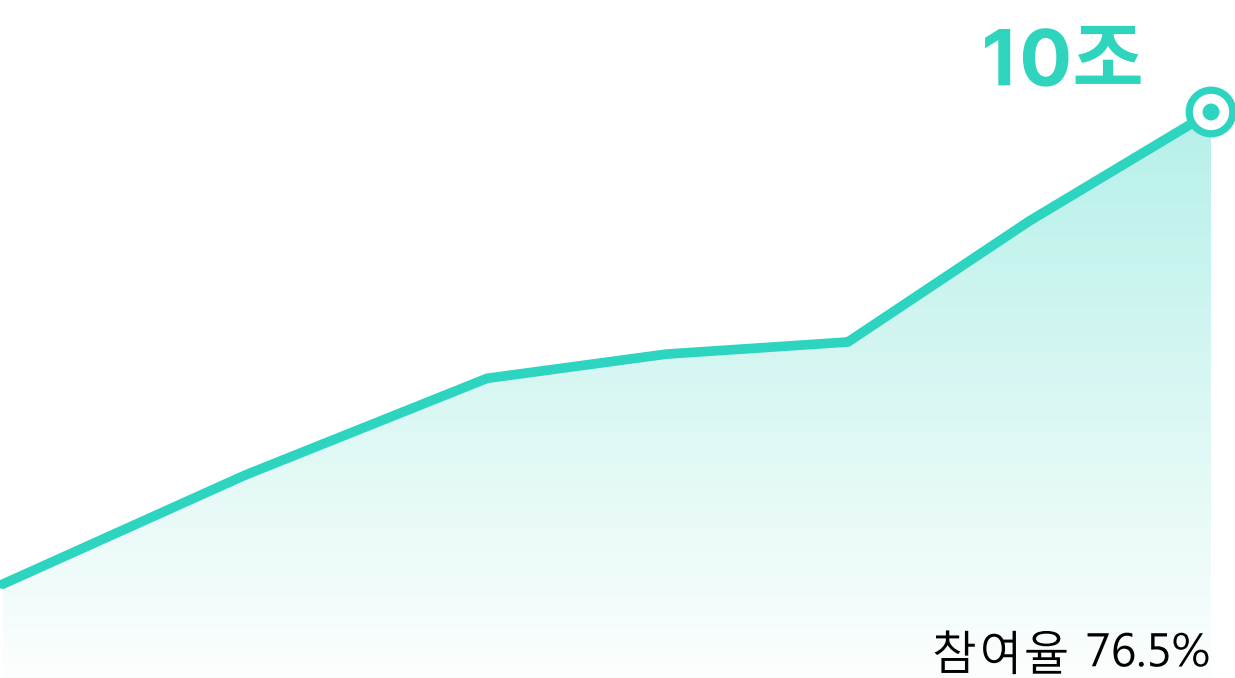


3조 227억

청년 자산형성 정책 예산,
미래적금 기여금 3% → 6~12%

수요

상승 꺾



89.84조

청년 연간 저축시장,
참여율 76.5% · 파킹통장 잔액 10조 돌파

01 시장 조사

청년 저축상품 지금 제대로 선택하면, 3년 간 200만원 넘게 더 받습니다.

정부 기여금

최대 216만 원

청년미래적금, 월 50만 원씩 3년 납입 시

산출: 납입원금 1,800만 원 × 기여금 매칭 12% 구간 기준
(구간별 6~12%)

이자 차이

연 약 10만 원

동일한 월 50만 원 적금이라도, 어떤 상품을 고르냐에 따라
1년 이자가 10만원 차이

산출: 월 50만 원 × 1년 적금,
금리 3%p 차이 적용 시 세전 이자 차이 약 9.8만 원

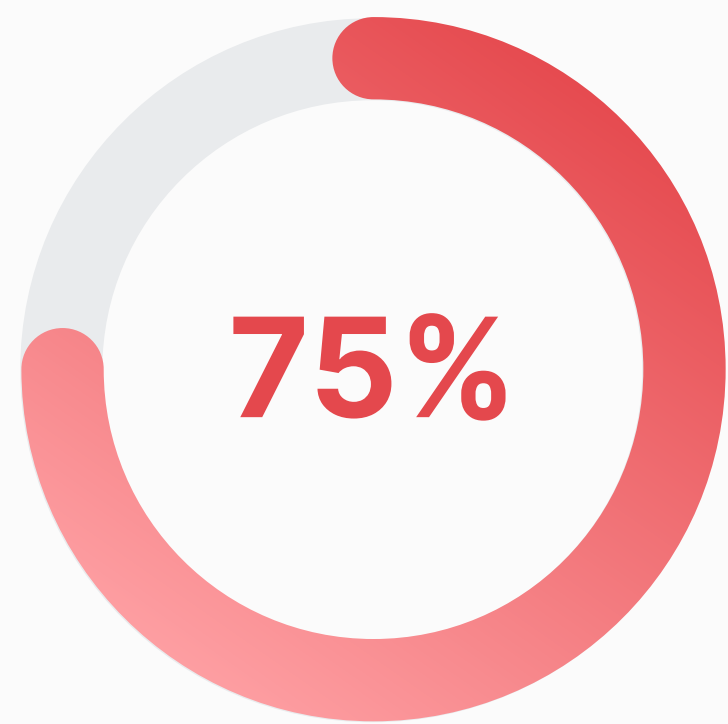
선택지는 398개입니다. 이 중 무엇이 '내 것'인지를 판단해야 합니다.

01 문제 정의

그런데, 청년들의 가입을 가로막는 **3개의 벽**이 있습니다.

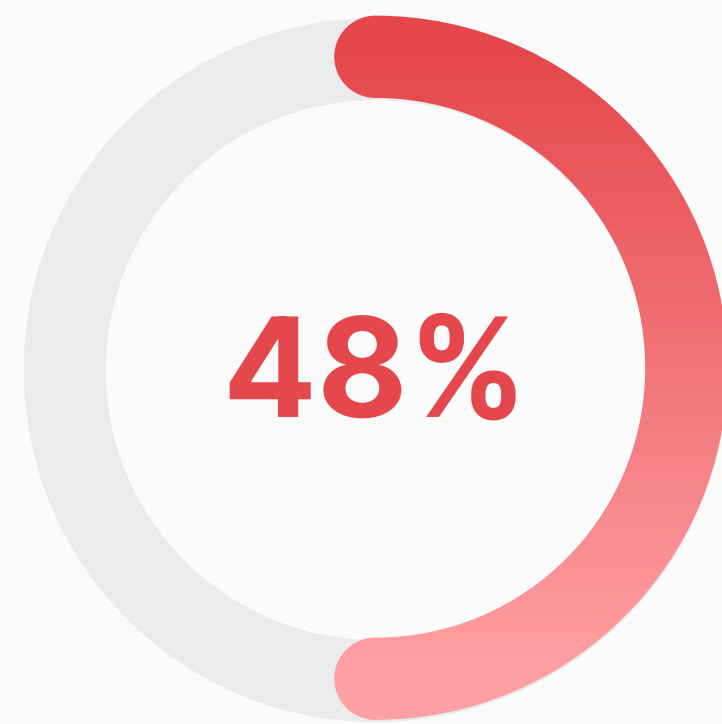
Y-Fin. 1차 설문조사 · 19~34세 청년 61명 · 14문항 · 2026.04 실시

1 정보 과부하 · 분산



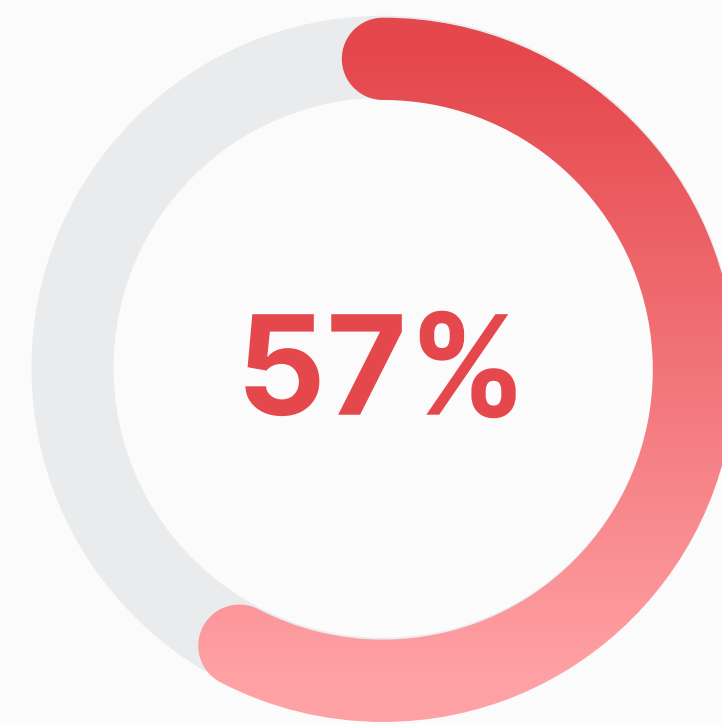
정부·은행 상품이 채널마다 흩어져 있고
토스·카카오페이조차 미통합,
정보 경로도 가족·커뮤니티 등 분산

2 우대금리 불투명



광고 최고금리와 실제 달성 금리가
평균 1~3%p 차이, 우대조건 모두
충족해야만 적용

3 금융 용어 장벽



20대 금융이해력 점수 전 연령 최저,
상품 약관 완독률 5%

결과

저축상품 비교만 하다가 청년 **64%**가 이탈하고, 가입한 뒤에도 **26%**가 중도에 해지합니다.

Y-Fin. 설문조사 이탈률 응답 · 청년도약계좌 중도해지율 (26.5)

02

솔루션 및 차별점

— 3단계 추천 로직 · 적합도 / 세후 실수령액 · 포지셔닝

02 솔루션

금융초보 청년을 위한
개인 맞춤 저축상품 추천 서비스, Y-Fin.

Y-Fin.의 핵심 목표

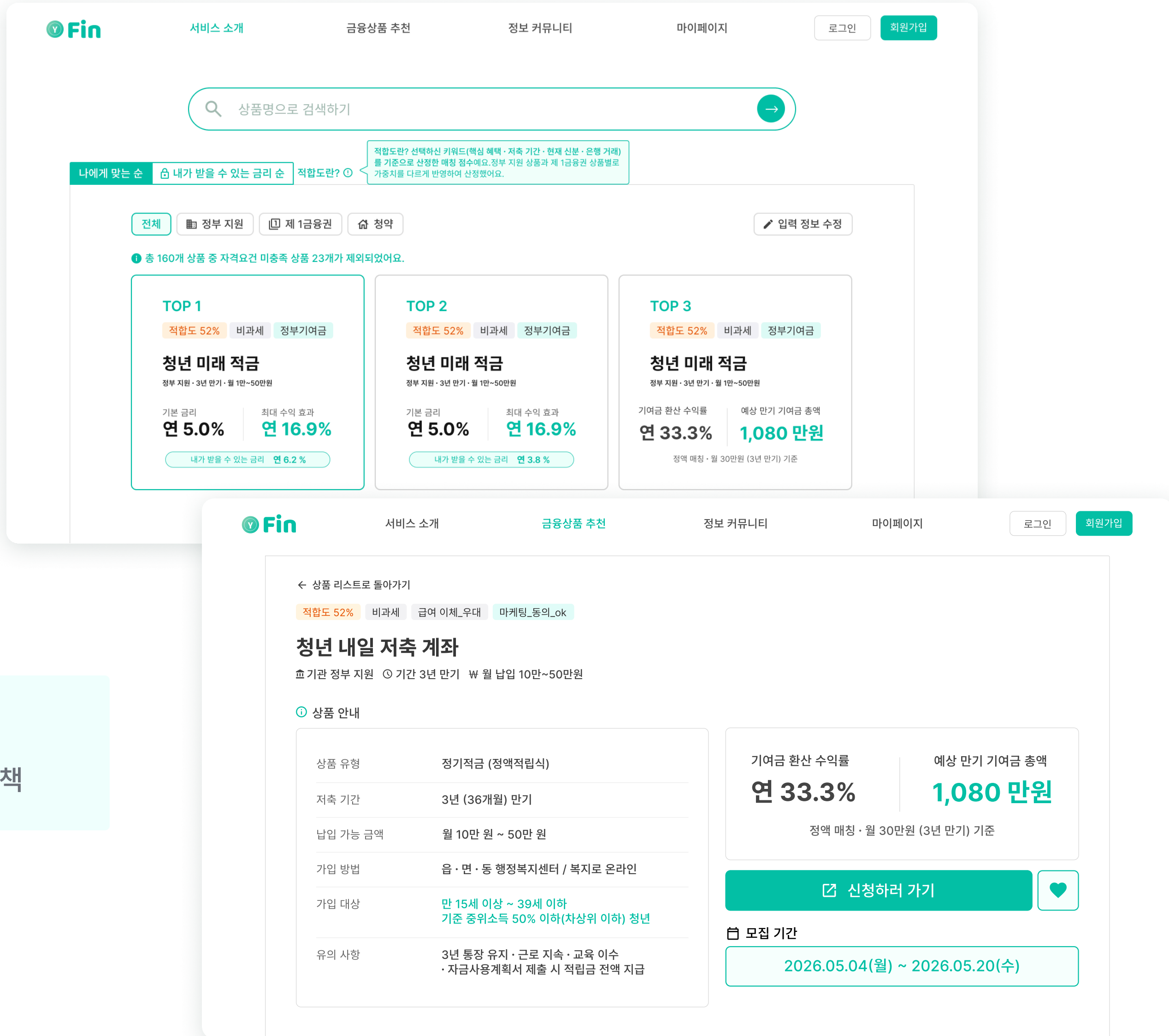
청년들의 저축상품 탐색-비교 과정을
자격요건 필터링과 개인 맞춤 정렬로 간편하게 만들고,
실제 가입 신청까지 이끌어내는 것

단기에치
파킹통장 · 초단기에적금

목돈만들기
예적금 · 정책저축

내집마련
주택청약 · 주거정책

Y-Fin. 최종발표회



02 솔루션

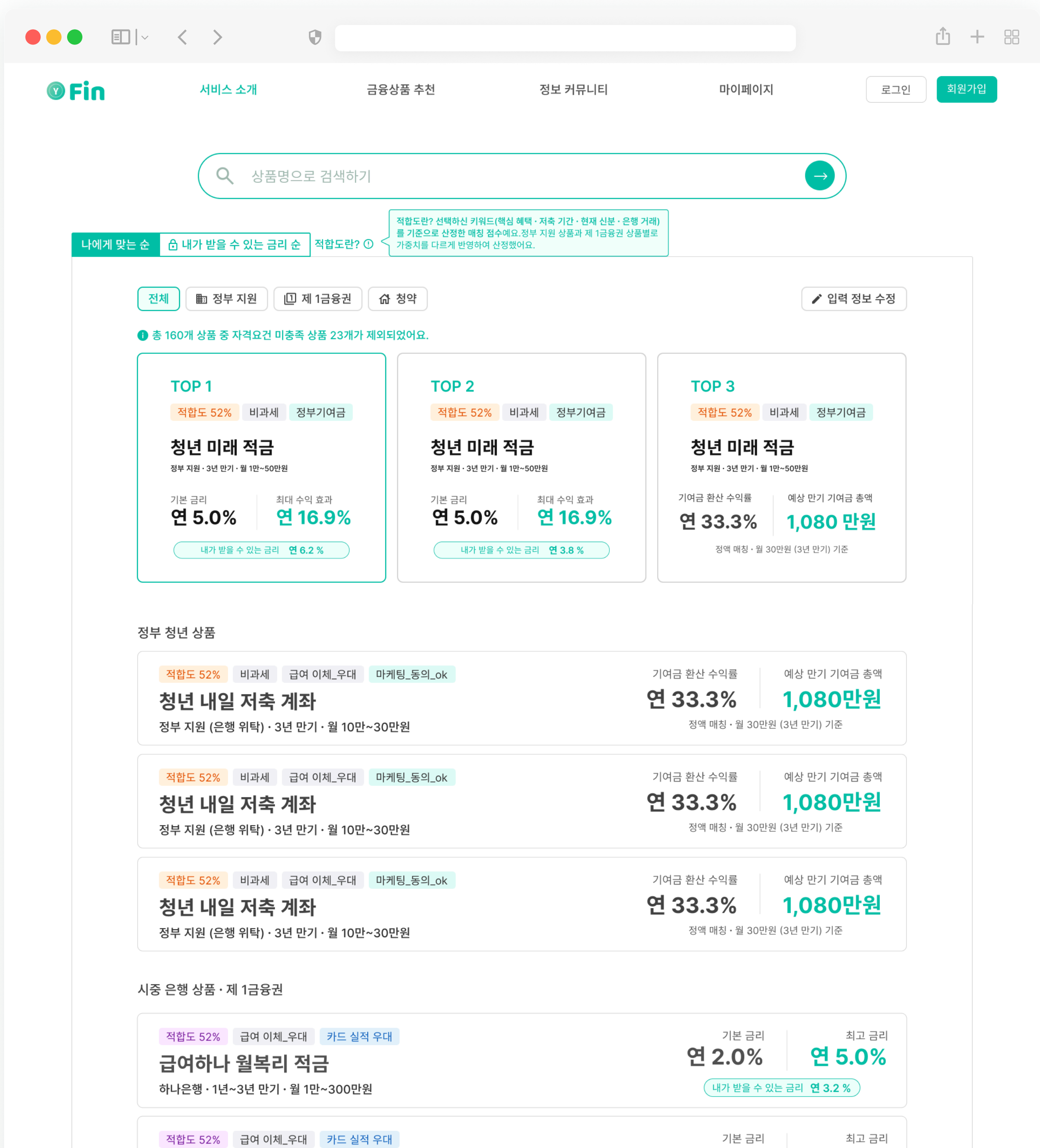
Y-Fin. 은 398개 상품 중,
청년 개개인에게 **딱 맞는 것만** 남깁니다.

1 모으고
정부와 은행의 상품을 한곳에 모으고

2 거르고
가입할 수 없는 상품은
내 자격 기준으로 거르고

3 정렬합니다
적합도 순 / 세후 실수령액 순으로
남은 상품을 정렬합니다

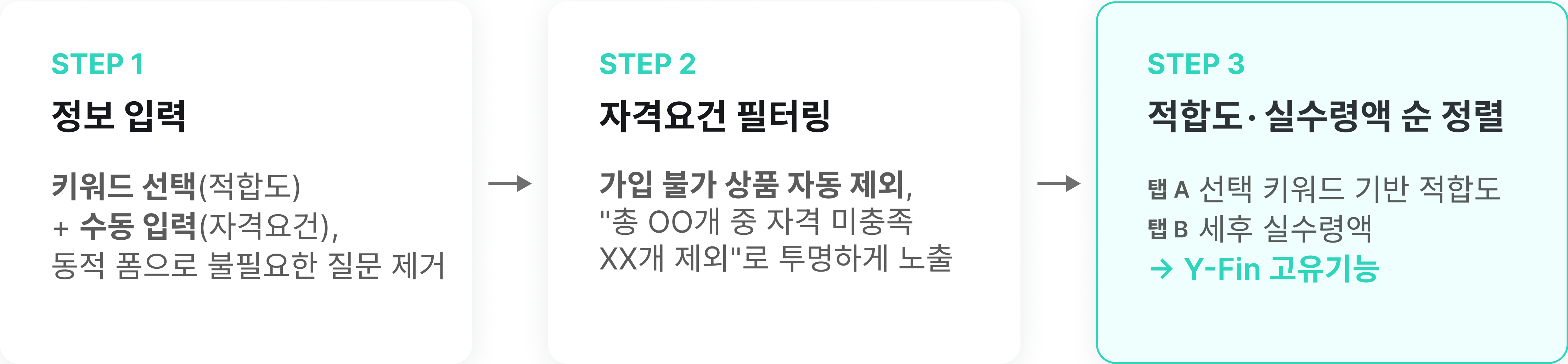
Y-Fin. 최종발표회



02 3단계 추천 로직

정보 입력 - 필터링 - 정렬

단 3단계 로직으로 청년들의 가입 고민을 해결합니다.



02 적합도

탭 A. 키워드만 선택하면, **나에게 맞는 적합도**로 알려드립니다.

유저가 선택할 키워드 화면

Q

→

STEP1

기본 정보
몇 가지 간단한 키워드 태그로 당신에게 Fin. 한 상품을 찾아드립니다.

핵심 혜택 (복수 선택)
#최고이율 중심 #비과세 #우대조건 간편 #정부기여금

은행 거래* (복수 선택)
필수 입력 항목입니다.
#자동이체_우대 #카드실적_우대 #급여이체_우대 #마케팅_동의_OK

이전

다음 단계

적합도 점수는 이렇게 만들어집니다

- 1 유저가 선택한 키워드와 상품 속성을 매칭해 100점 만점으로 계산합니다
- 2 항목별 가중치는 상품 데이터 특성을 반영하여 도출하였습니다 (고특성 30점, 일반 20점)
- 3 정부 상품과 은행 상품은 구조가 달라, 배점 체계를 분리해 각각 계산합니다

02 세후 실수령액

탭 B. 광고 속 금리가 아니라, **내가 받을 수 있는 실수령액**을 보여드립니다.



BEFORE

비교 자체가 불가능

정부 상품

금리 미공시 → 정부 상품끼리 수익률 비교 불가

은행 상품

광고 최고금리와 실제 금리 **평균 1~3%p 괴리**,
같은 금리라도 예금이 적금보다 **실수령 약 1.8배 ↑**

설문조사 비교 중요요소 1위 '금리 수준 84%'에 직접 대응



AFTER · Y-Fin 산식

상품 유형별로 같은 기준을 세워 비교

정부 상품

기여금 → **연환산 수익률 산식**으로 변환,
정부 상품 간 수익률 비교 가능

은행 상품

기본금리 + **실제 충족 가능한 우대금리**만 합산한
세후 실수령액 기준 정렬

정렬에 광고비 절대 미반영

모두가 반쪽만 채운 자리, Y-Fin이 유일합니다.

포지셔닝 맵

상품통합
↑
상품단일

온통청년 · 금감원 파인

통합 범위는 넓으나,
개인화 부재

Y-Fin.

정부·은행 통합
+ 청년 맞춤

SOL · NH올원뱅킹

자행 상품 편향,
개인화 부재

토스 ·뱅크샐러드

개인화는 일부,
정책상품 부족

개인화 수준 범용 → 청년 맞춤

기능별 비교 ○ 제공 · △ 부분 · X 미제공

핵심 기능	Y-Fin	공공	핀테크	은행앱
정부 · 은행 상품 통합	○	○	△	X
자격요건 자동 필터링	○	X	△	X
적합도 순 정렬	○	X	X	X
실수령액 순 정렬	○	X	X	X
전역 금융 용어 풀이	○	X	△	X

적합도 · 실수령액 순 정렬은 Y-Fin만의 고유 기능

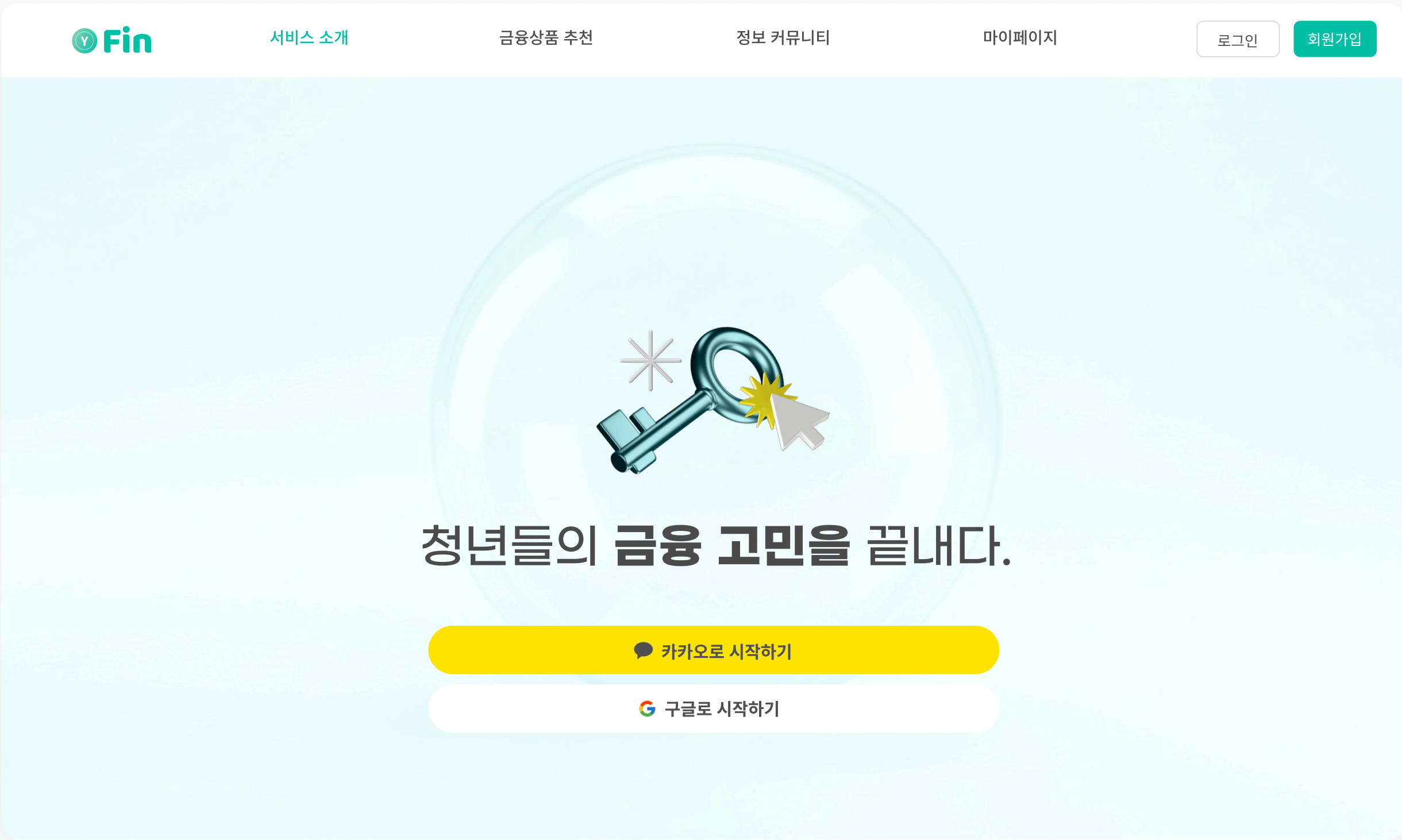
Y-Fin. 최종발표회

03

서비스 진행 현황

— 서비스 시연 영상 · 개발 현황 · AI 활용

03 서비스 시연 영상

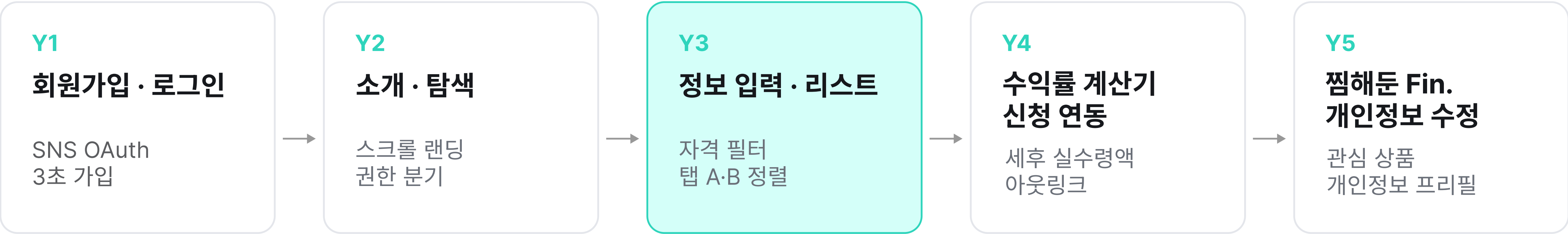


▶ 서비스 시연 영상

로그인 → 정보 입력 → 리스트 → 상세정보 → 수익률 계산기 → 마이페이지

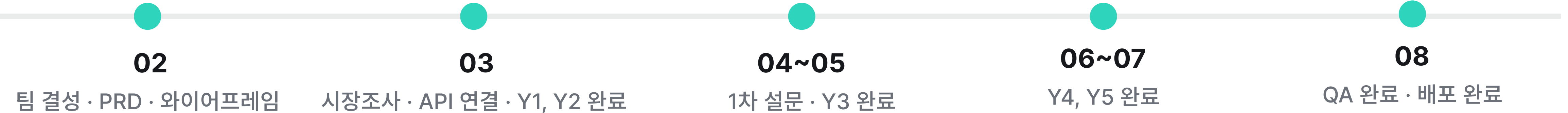
03 진행 현황

방금 보신 화면들, 전부 **실제 동작** 중인 기능입니다.



MVP P0 전 기능 개발 완료 · 내부 QA 완료 · 배포 완료

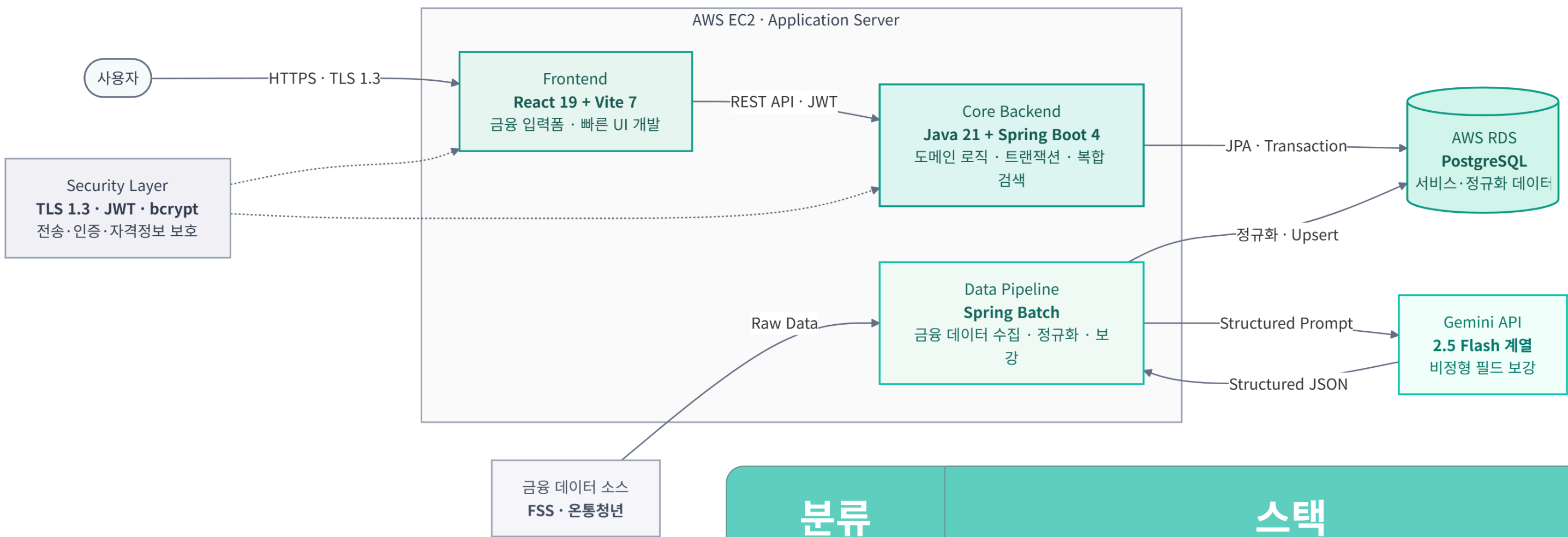
Y-Fin. 26.02~08 마일스톤



Y-Fin. 최종발표회

03 진행 현황

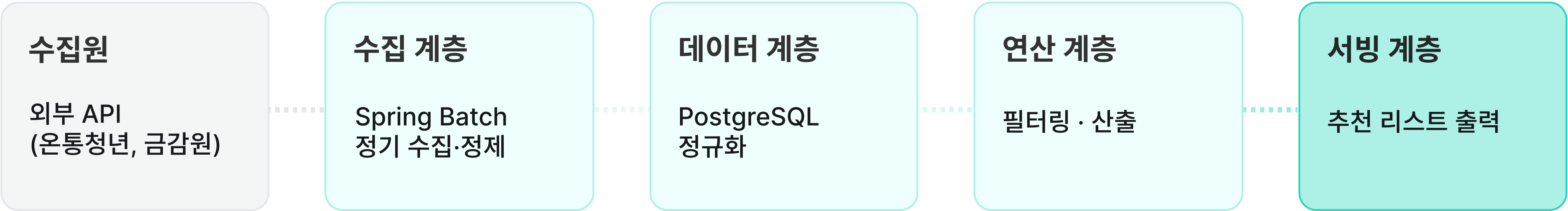
검증된 스택과 명확한 역할 분담으로, 정식 런칭까지 완주합니다.



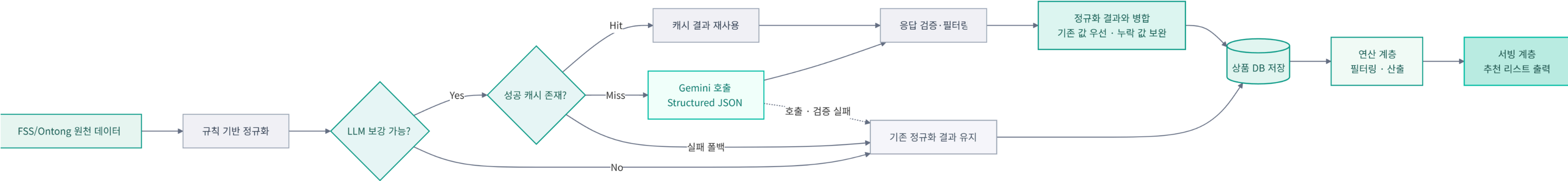
분류	스택	선정 근거
FE	React, Vite	복잡한 금융 입력폼 상태 관리
BE	Java, Spring Boot, PostgreSQL	도메인 로직 분리, 트랜잭션 관리, 복합 조건 검색
인프라	AWS EC2, RDS	애플리케이션/DB 분리 운영
AI	Gemini API (2.5 Flash lite)	데이터 파이프라인 비정형 필드 보강
보안	HTTPS/TLS 1.3, JWT, bcrypt	금융 서비스 신뢰도 확보

03 진행 현황

추천 결과에 오류가 없도록, 체계적으로 데이터 구조를 설계했습니다.



수집·가공·서빙 계층을 분리했습니다.
대출·마이데이터 API가 추가돼도 이 구조는 바뀌지 않습니다.



03 진행 현황

LLM과 규칙 기반 처리를 결합해 위험 요인을 해소하였습니다.

```
@Override
public ProductDraft enrich(ProductRaw rawProduct, ProductDraft draft) {
    if (!enabled() || !draft.shouldSaveProduct()) {
        return draft;
    }

    LlmProviderClient providerClient = providerClient();
    String prompt = promptBuilder.build(rawProduct, draft);
    String requestHash = Sha256.hex(prompt);
    LlmEnrichmentCache cache = cacheStore.findOrCreate(rawProduct, requestHash);

    if (cache.getStatus() == LlmEnrichmentCacheStatus.SUCCESS
        && cache.getResponseJson() != null
        && requestHash.equals(cache.getRequestHash())) {
        cacheHits.incrementAndGet();
        return fromCache(cache, rawProduct, draft);
    }
    if (cache.isFailedRetryBlocked(Instant.now(), FAILED_RETRY_COOLDOWN)) {
        cooldownSkips.incrementAndGet();
        log.debug(
            format("Skipping FSS LLM enrichment during failed retry cooldown. rawId={}, externalId={}, failureCount={}",
                rawProduct.getId(),
                rawProduct.getExternalId(),
                cache.getFailureCount()
            )
        );
        return draft;
    }

    try {
        llmCalls.incrementAndGet();
        LlmProductEnrichment enrichment = providerClient.enrich(new LlmProductEnrichmentRequest(
            properties.llm().model(),
            prompt,
            properties.llm().schemaVersion()
        ));
        validator.validate(enrichment);

        cacheStore.saveSuccess(cache, requestHash, enrichment);
        return merger.merge(rawProduct, draft, enrichment);
    } catch (Exception e) {
        llmFailures.incrementAndGet();
        log.warn(format("FSS LLM enrichment failed. rawId={}, externalId={}", rawProduct.getId(), rawProduct.getExternalId(), e));
        cacheStore.saveFailed(cache, requestHash, truncate(e.getMessage()));
        return draft;
    }
}
```

데이터 수집 후 LLM 코드 보강

규칙 기반으로 추출이 어려운 비정형 텍스트 필드에 LLM 적용
캐시 검증·재시도 정책으로 효율성 향상

```
@Component
public class RegionKeywordRecognizer implements KeywordRecognizer {

    @Override
    public List<KeywordValueEnum> recognize(ProductDraft productDraft, ProductPropertyDraft propertyDraft) {

        String providerName = propertyDraft.providerName();
        Set<KeywordValueEnum> keywords = new HashSet<>();
        if (providerName != null && !providerName.contains(s: "은행")) {
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_SEOUL, ...tokens: "서울");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_BUSAN, ...tokens: "부산");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_DAEGU, ...tokens: "대구");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_INCHEON, ...tokens: "인천");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_GWANGJU, ...tokens: "광주");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_DAEJEON, ...tokens: "대전");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_ULSAN, ...tokens: "울산");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_SEJONG, ...tokens: "세종");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_GYEONGGI, ...tokens: "경기");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_GANGWON, ...tokens: "강원");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_CHUNGBUK, ...tokens: "충북", "충청북도");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_CHUNGNAM, ...tokens: "충남", "충청남도");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_JEONBUK, ...tokens: "전북", "전라북도");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_JEONNAM, ...tokens: "전남", "전라남도");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_GYEONGBUK, ...tokens: "경북", "경상북도");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_GYEONGNAM, ...tokens: "경남", "경상남도");
            addIfContains(keywords, providerName, KeywordValueEnum.REGION_JEJU, ...tokens: "제주");
        }

        if (!keywords.isEmpty())
            return keywords.stream().toList();

        return List.of();
        // return List.of(...
    }
}
```

규칙 기반 (Rule-based) 데이터 처리

규칙이 명확한 데이터는 코드로 추출해 비용 절감 및 정확도 향상
LLM과 규칙 기반의 역할 분리 원칙

03 진행 현황

AI를 개발 보조 도구로 활용해, 핵심 기능 구현에 집중했습니다.

파트	도구	활용 목적
기획	Claude	필터링-정렬 로직 초기 설계 · 우대금리 조건의 입력 단계별 반영
디자인	Claude	와이어프레임 기반 초기 설계 · 컬러 시스템 구축
백엔드	Codex/Claude	DTO·Entity·Repository 템플릿 · 테스트 코드 초안
프론트엔드	Claude	컴포넌트 구조 초안 · 반복 UI 정리 · UX 개선
데이터	Codex/Claude	온통청년·금감원 API 응답 분석 · 정규화 규칙 설계
배포	Codex	테스트 배포 시 EC2·Docker Compose 스크립트 및 환경 설정 검토

LLM으로 완성한 코드는 커밋/PR 시 리뷰한 후 반영
API Key·DB 비밀번호 등 민감정보 미입력

04

서비스 향후 계획

— 상품군 / 기능 확장 · 시장 진입 로드맵 · 정량 목표(KPI)

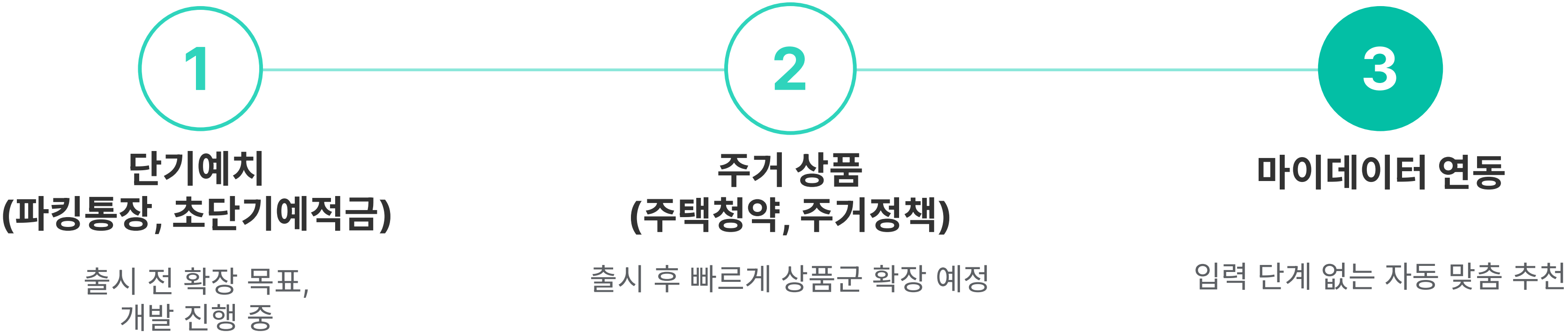
04 상품군 / 기능 확장

저축상품 추천에서 멈추지 않고, **청년 금융 슈퍼앱**으로 나아가겠습니다.

상품군 / 기능 2축으로 확장

01 상품군 확장

청년들의 선택지를 넓힙니다.



02 기능 확장

청년들이 돌아올 이유를 만듭니다.

청년 금융뉴스 · 커뮤니티

청년 금융 정보의 **44.4%**는 커뮤니티에서 오갑니다.
그 대화가 **정확한 정보** 위에서 이루어지도록 합니다.

정부 상품 캘린더 · 일정 알림

가입 기간이 정해진 **정부상품**의 캘린더뷰 및
신청 일정 알림 제공으로 리텐션을 높입니다.

PMF 먼저, 수익화는 나중에 진행합니다.

PHASE 1 · 진입

LAM · 베타 100명

26.06~09 (4개월)

- 상품군 / 기능 확장
- 베타 테스트 진행
- MVP 버전 출시
- SW 저작 등록

PHASE 2 · 안착

SOM 진입 · 누적 5천 명

26.10~27.06 (9개월)

- 본격 마케팅 진행
- 광고 BM 도입
(B2B, 추천 영역과 격리)
- MAU 1,000 목표

PHASE 3 · 확장

SOM 달성 · 누적 5만 명

27.07~29.06 (2년)

- 마이데이터 연동
- 구독 BM 도입 (B2C)
- 데이터 BM 도입 (B2G, B2B)
- MAU 5,000 목표

04 정량 목표 (KPI)

다짐이 아닌, 실제 수치로 기대 효과를 증명하겠습니다.

<div>80%</div> <div>정보 입력 완료율</div> <div>Primary · 퍼널 분석</div>	<div>15%</div> <div>실제 신청률</div> <div>아웃링크 클릭률</div>	<div>7%</div> <div>7일 리텐션</div> <div>코호트 분석</div>	<div>5천</div> <div>MAU</div> <div>3년 내 · GA4</div>
--	--	---	--

금융위 2026 정책 기조와 연계하여, 사회적 가치를 실현합니다.

정책 효과성 제고

자격을 몰라서 놓치는 정부 사업의
실제 가입률을 높임

청년미래적금 예산 약 86만 명분 미신청,
정책은 있으나 도달하지 못함

포용금융 실현

금융 정보 비대칭 해소,
다크패턴 방지

광고금리 아닌 세후 실수령액으로
금리 오인 가입 차단

신뢰 보호 및 중립 유지

추천 정렬에 광고비 미반영,
광고는 별도 영역 격리 및 명시

판매·중개 아닌 정보 제공 서비스

FIN
FOR
YOU!

PO 이서빈

BE 강지원, 최유렬

PD 정예담

FE 안지원